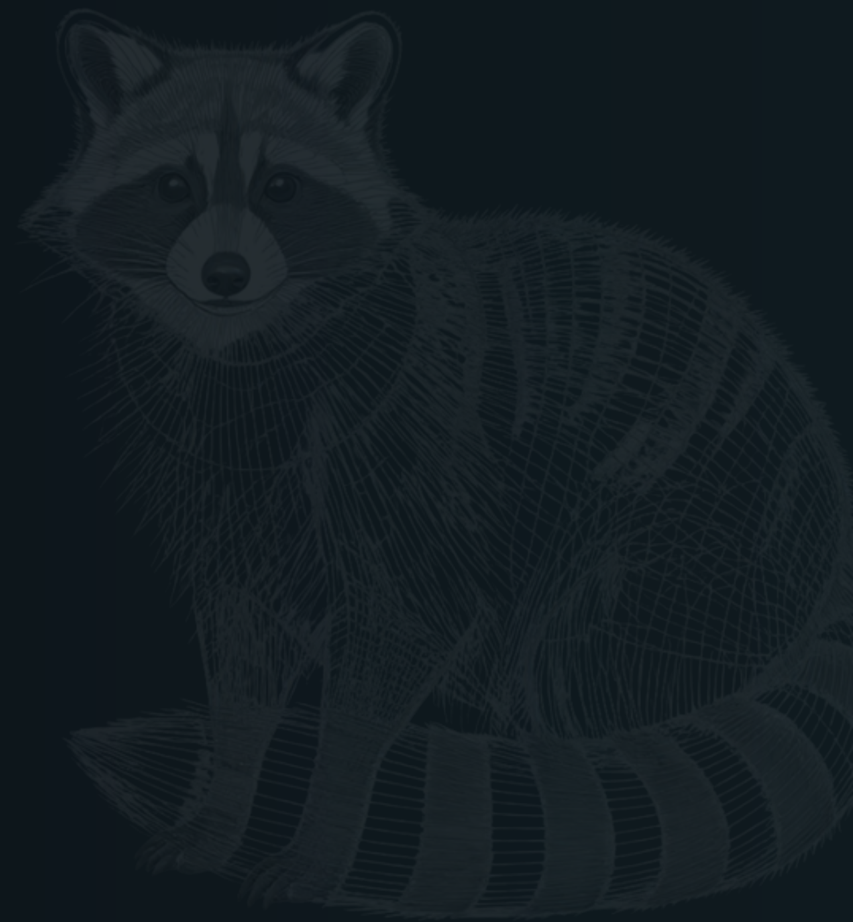
Ubuntu 26.04 LTS Server Considerations

～26.04 LTSに乗り換える？　乗り換えない？～

おしながき：

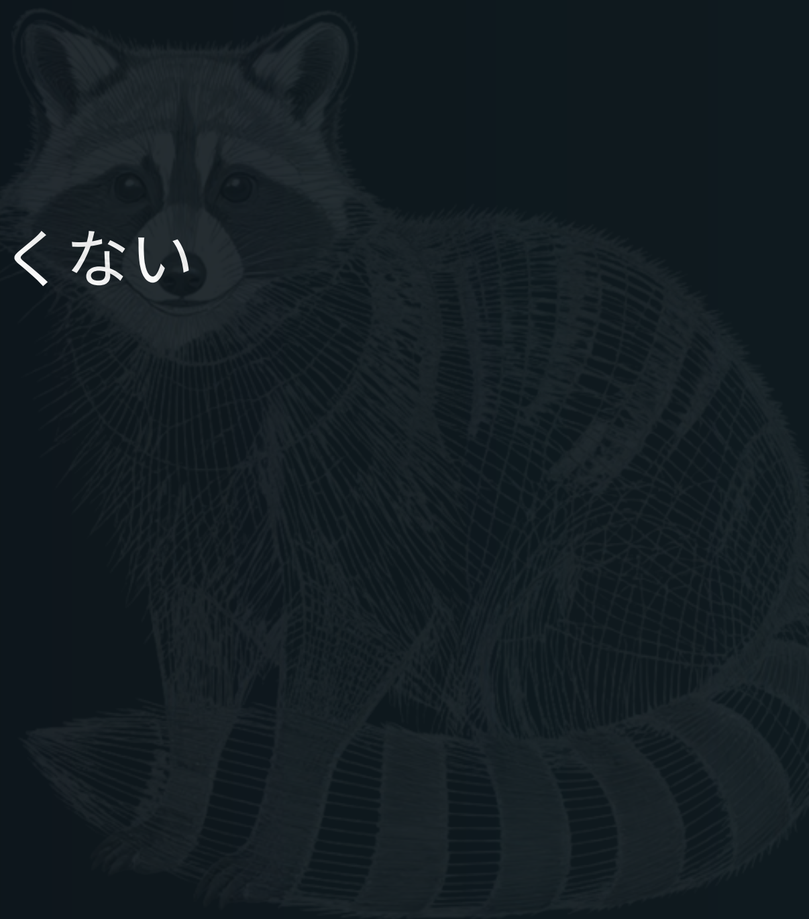
- 26.04 LTS Serverの新機能のポイント
- アップグレードする, しないの基準を考える
- 今やるべきことの整理



想定対象：
22.04 LTSをサーバー用途で使っている方
（18.04 LTS、20.04 LTSのUbuntu Pro課金勢も含む）

サブの想定：
Ubuntu以外のLinuxディストリビューションをサーバー用途で使っている方

まず結論：



- uutils由来の（Rust版）coreutilsはそこまでヤバくない
- 他の観点でもそこまでヤバくない
- ただ、他にやることがあるかもしれない

という方向性の結果として↓

想定対象：
22.04 LTSをサーバー用途で使っている方
（18.04 LTS、20.04 LTSのUbuntu Pro課金勢も含む）

22.04 LTSを使っている場合：

選択肢：

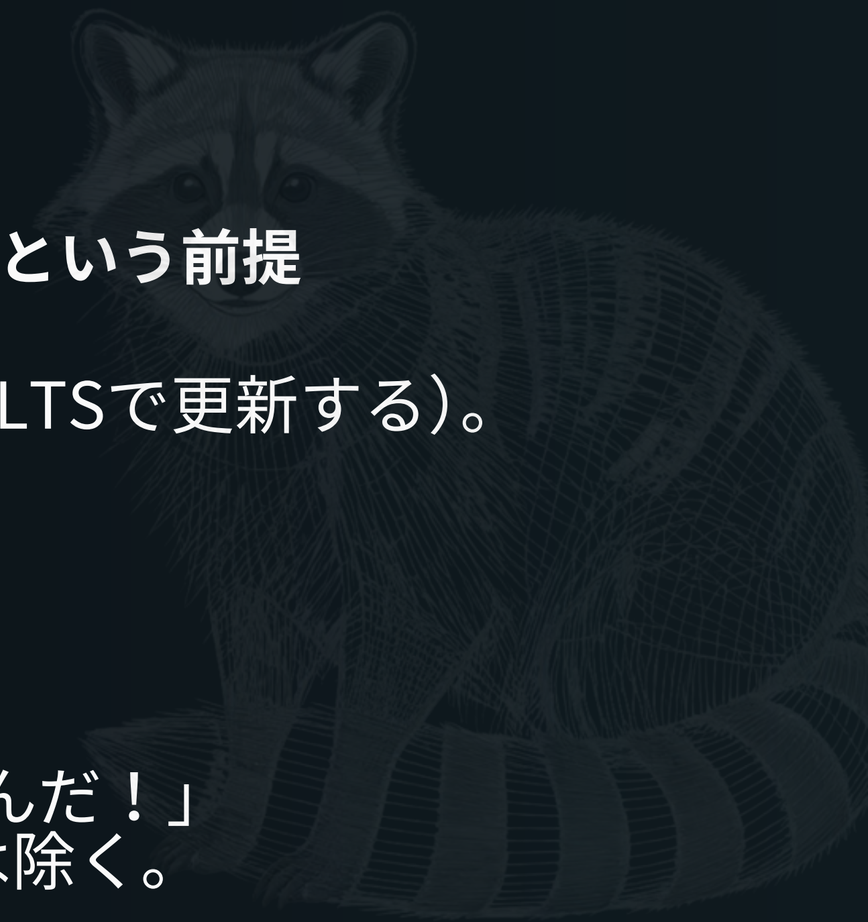
A. 24.04 LTSに更新する。

B. 26.04 LTSに更新する。

C. 更新せずUbuntu Pro調達の準備をする。

D. 別のディストリビューションやOSにする。

24.04 LTSを使っている場合：

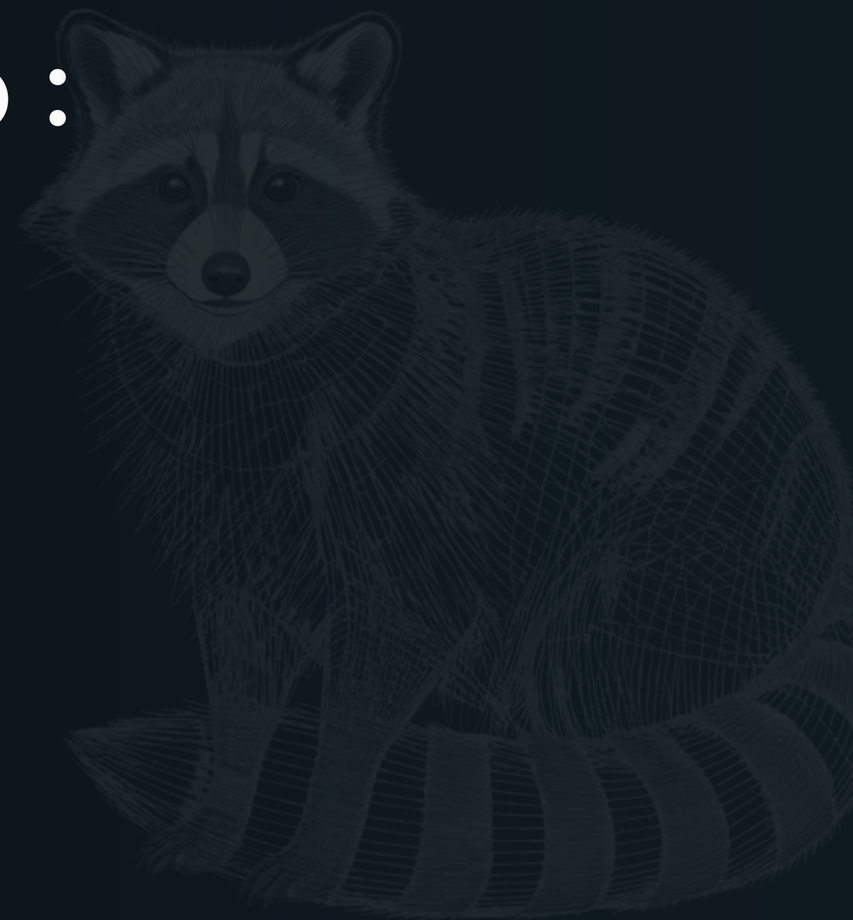


※ 今年〜来年、たぶんAI関連で超忙しくなりません？　という前提

- とりあえず置いておいても良さそう（＝ 28.04 LTSで更新する）。
  - 26.04 LTSへ積極的に更新するべき理由は**多分ない。**
  - 2028-29に超忙しくなることが見えているなら話は別。
    （その時期に終了するミドルウェアが大量にある等）
- 「使っているソフトウェアをAIで全部検査するんだ！」
  みたいな例外的なシチュエーションにある人は除く。
  - coreutilsを雑にスキャンすると、おそらく大量の偽陽性が……。

26.04 LTS Serverの新機能のポイント

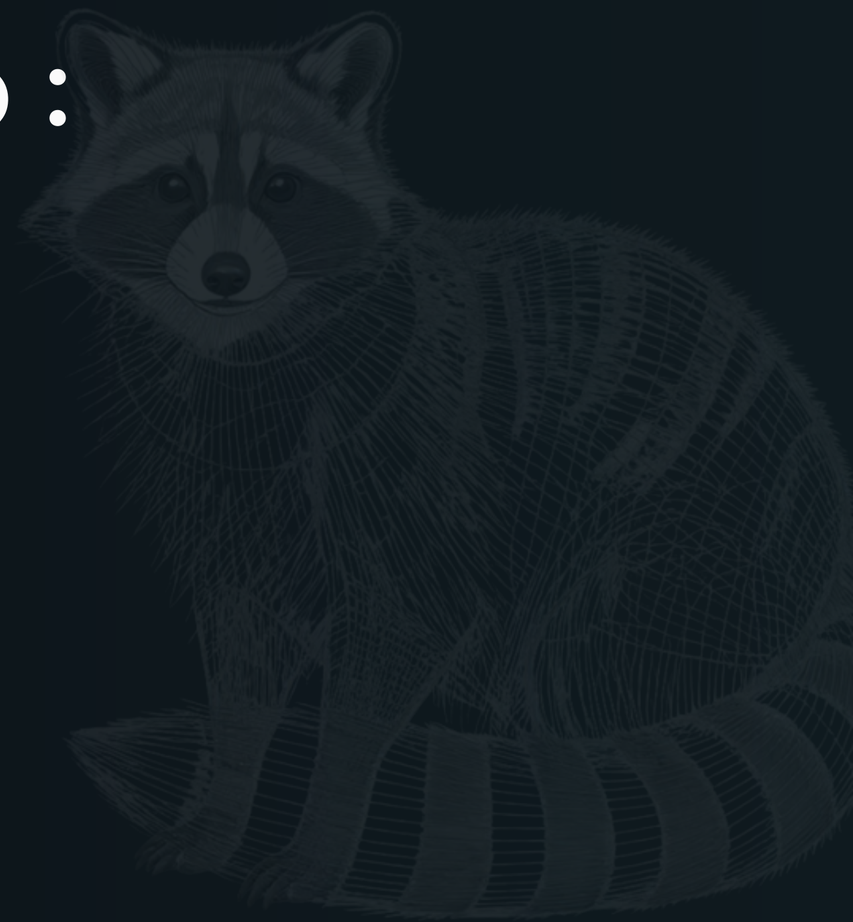
新機能のうちServerに影響しそうなもの：



- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

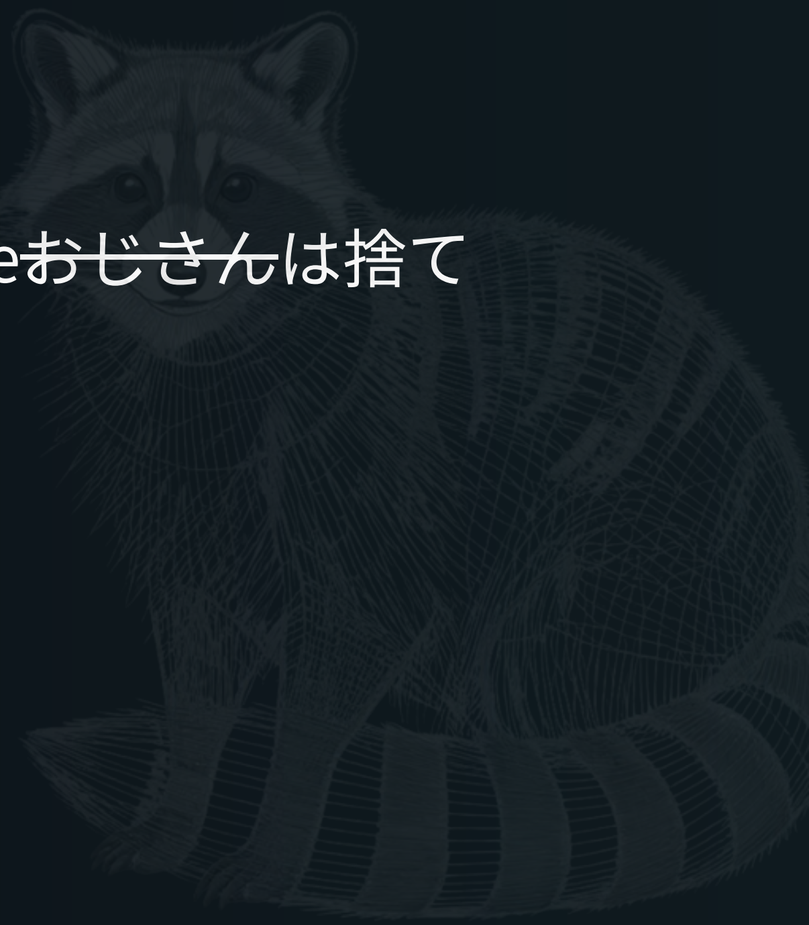
新機能のうちServerに影響しそうなもの：

- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

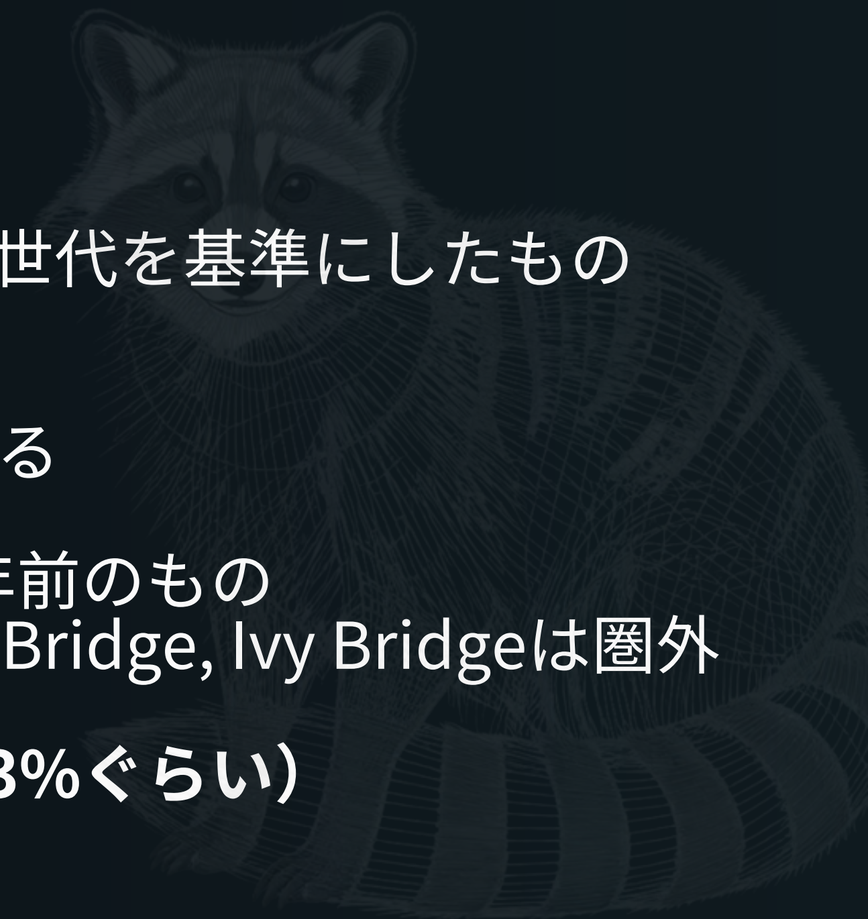


わかりやすくした話

- Ubuntu Server的には、Sandy Bridge, Ivy Bridge~~おじさん~~は捨て
- Haswell~~おじさん~~はまだしばらくOK



x86-64v3(AMD64v3) って何？



- CPUの命令セットのうち、Haswell/Excavator世代を基準にしたもの（AVX2やFMA3サポートが必須になる）
- 2013-2014年ぐらいのCPUならサポートしている

  逆に言うと「サポートしていない」CPUは12年前のもの
  「実用」範囲に入っているCPUの中ではSandy Bridge, Ivy Bridgeは圏外
- **ちょっと速くなる（平均で1%、うまくすると3%ぐらい）**

より詳しくは https://gihyo.jp/admin/clip/01/ubuntu-topics/202312/15 あたりを参照してください。

知っておくべき知識：CPU性能は落ちることがある

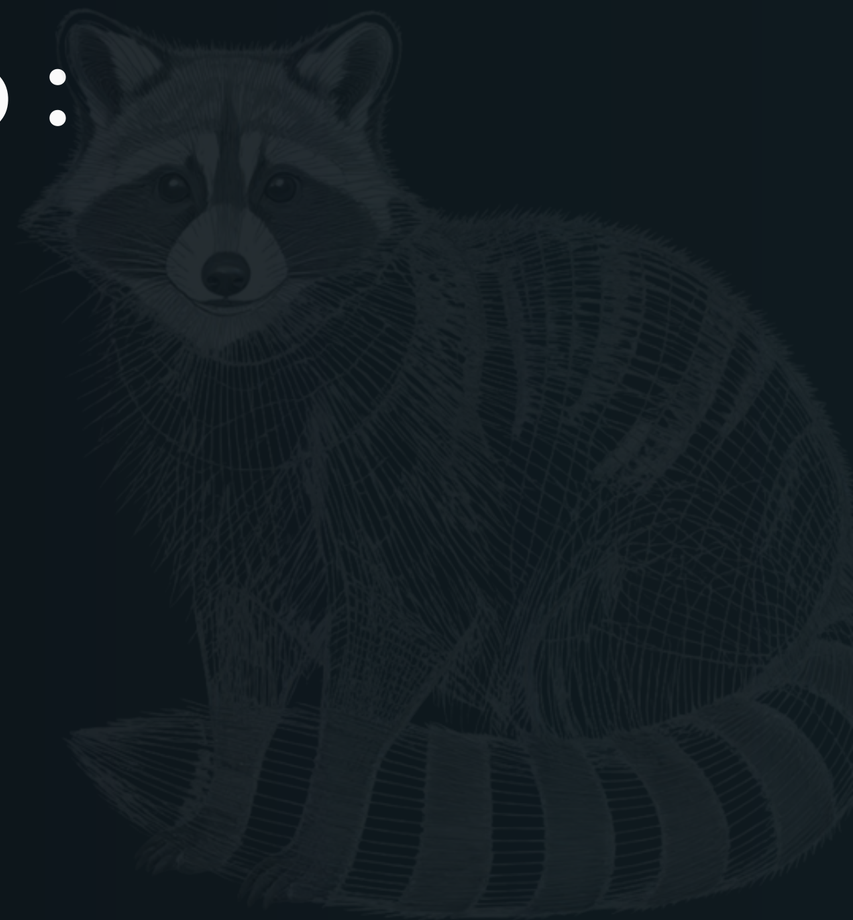
- CPU性能は稼働中に「減る」ことがありうる
  （主な原因：脆弱性対応＝マイクロコード更新による性能特性の変化）
- 各種のpacked演算（新命令群が対応しているもの）は使っておくに越したことはない
- AMD64v3への切り替えも進んでいる
  （とはいえ互換性を失わせてまで必須にするかというと……）
- この流れだと28.04 LTSではServerは全部AMD64v3前提では？　感
  古いハードウェアで動かしている場合にどうするべきかには影響しそう
  （禁句：動いているSandy Bridge or Ivy Bridgeなんていくやさんち以外にあるのか？）

各種クラウドイメージはAMD64v3がデフォルトに更新済

- 通常のServer ISOはまだ
- Google Cloudでは、N1の一部CPU搭載インスタンスのサポートが終了
  （Haswell-EP以前 = Sandy Bridge-EP/Ivy Bridge-EP搭載のものがダメ）
- AWSでは non-nitro なインスタンスのサポートが終了
  （XenベースのM1-M4, C1-C4, R3-4, I2, G3, P2, P3, P3dnがダメ）
  ※ Haswell-EP以前かどうかと関係ないのがポイント。

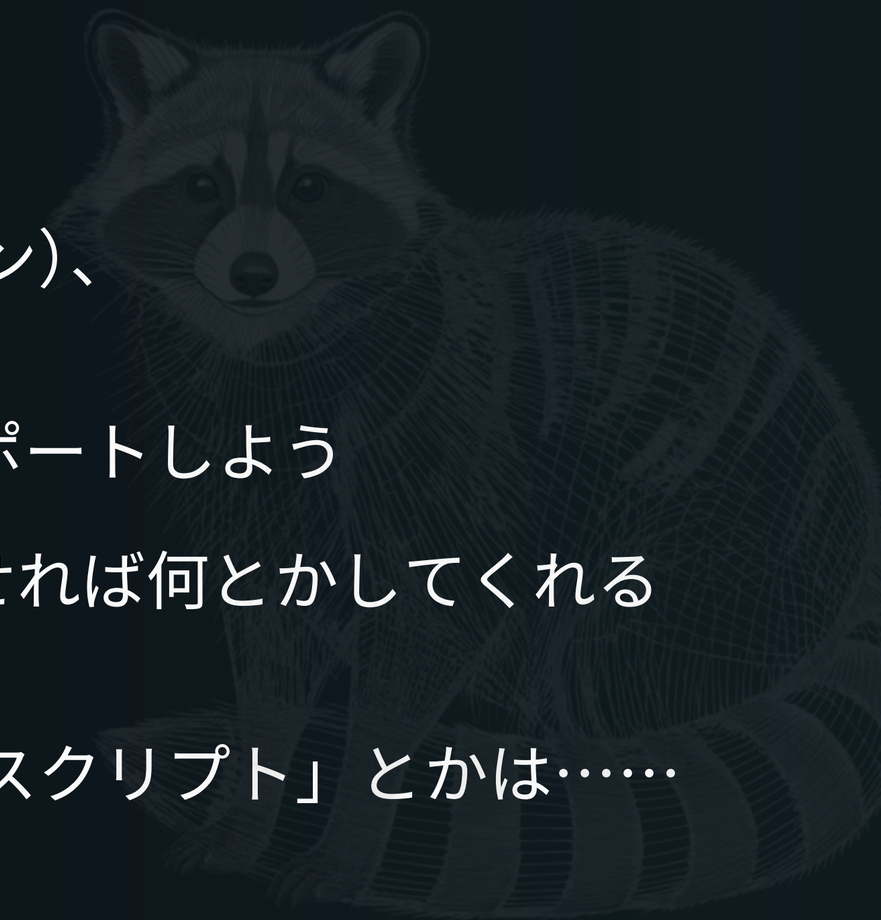
どちらにせよ、**引っ越せない理由がある場合はその理由は解決した方がいいです……。**

新機能のうちServerに影響しそうなもの：



- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

Rust版coreutils + sudo



- 結構いろいろ違う（特にマイナーなオプション）、
  しかし「違う」が致命的ではないはず？
- なにか見つけたら ubuntu-bug コマンドでレポートしよう
- ワークアラウンドの類はAIエージェントに任せれば何とかしてくれる
  （Codexの5.4 miniとかでたぶんOK）

  ※ ただし「どう動くべきなのか誰も知らないスクリプト」とかは……

- sudoのパスワード入力のエコーバックが気持ち悪い？

新機能のうちServerに影響しそうなもの：

- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

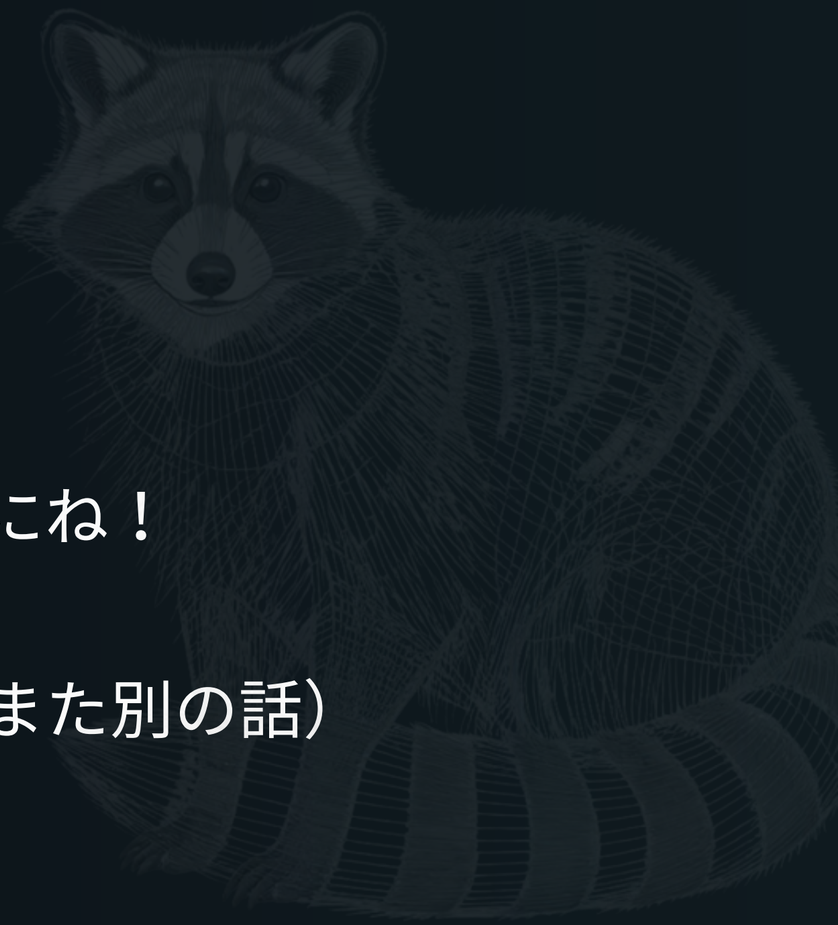
カーネル7.0とTDX, E-IOMMU, 仮想化スタックの更新

- XFSを使っている人は嬉しいかも
  他に大きな変更点があるかというと、多分あんまりない
- ただしTDXを使いたい（＝Confidential VMを動かす）場合と、
  GPGPUに激しく動いてもらう用事がある（＝Enhanced IOMMUが
  ほしい）場合は話は別
- 仮想化スタックがHWEで更新されるようになった点は将来のポイント
  （28.04までの各リリースの仮想化スタック更新を26.04でも受け取れる）

新機能のうちServerに影響しそうなもの：

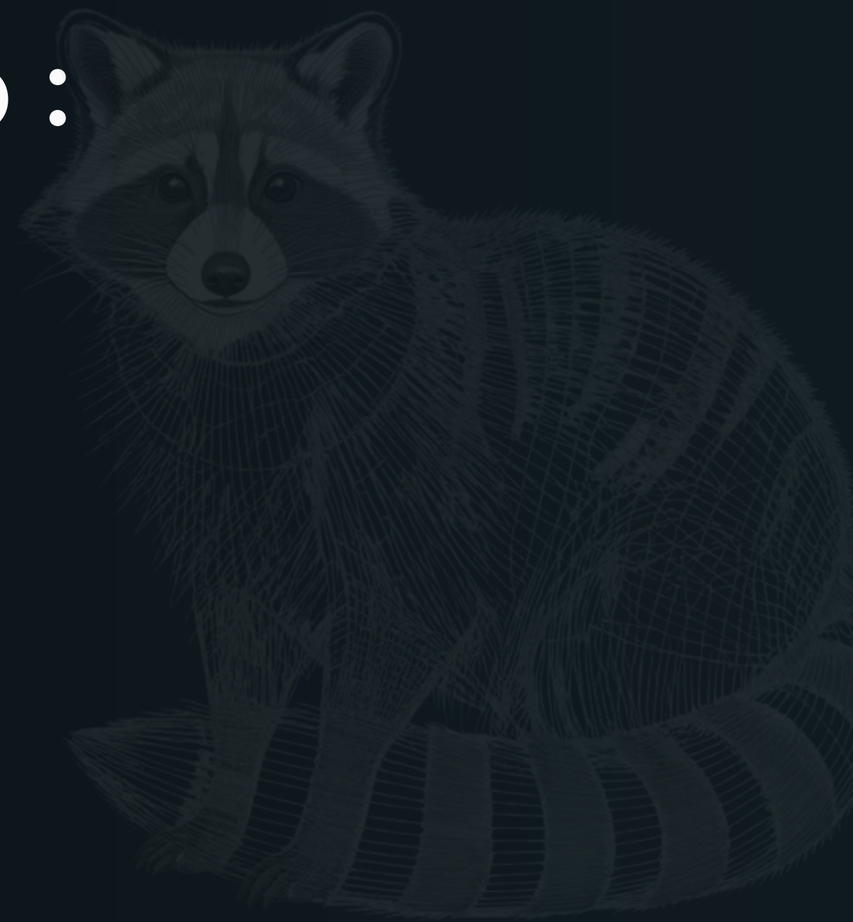
- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

各種ソフトウェアの更新



- PostgreSQL 18とDocumentDBの追加は大きい
  （用がない人にとってはどうでもいい）
- NTPd to Chrony
  設定方法がちょっと変わるのでハマらないようにね！
- あとはまあ……いつもの……？（PQCまわりはまた別の話）

新機能のうちServerに影響しそうなもの：



- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

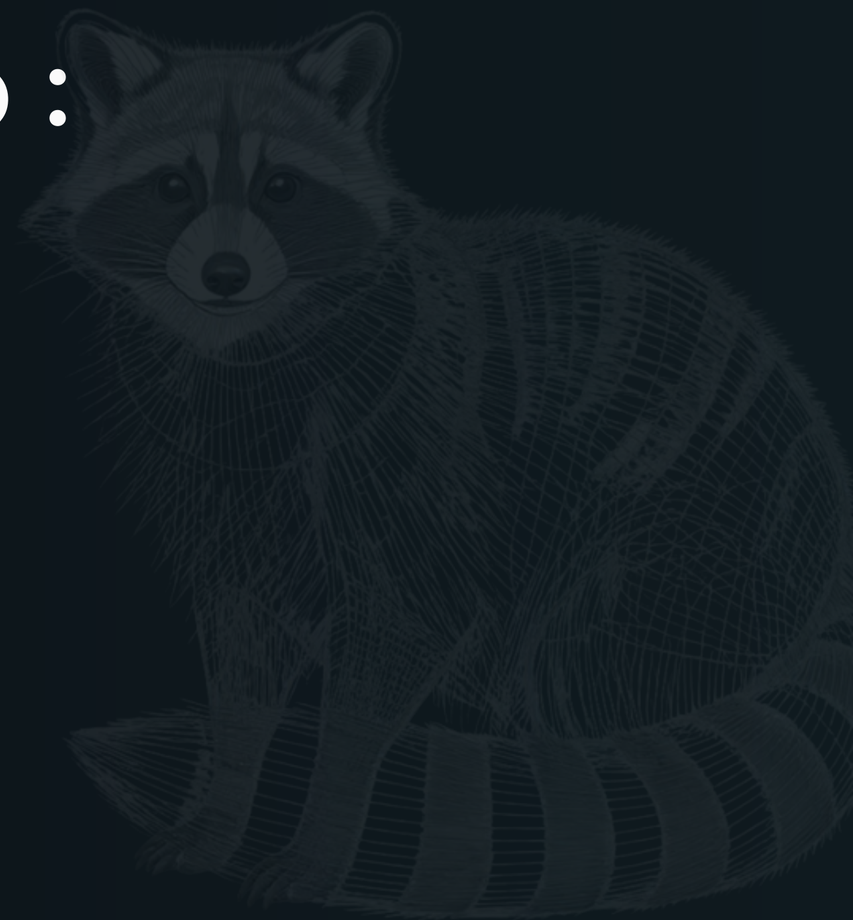
CUDA, ROCmがUbuntuのリポジトリ入り

Before: 自分でリポジトリを追加する（そしてリポジトリがたまにoutdatedになる）

After: 「何も考えずにaptで足す」でインストールできるし、アップデートもOK

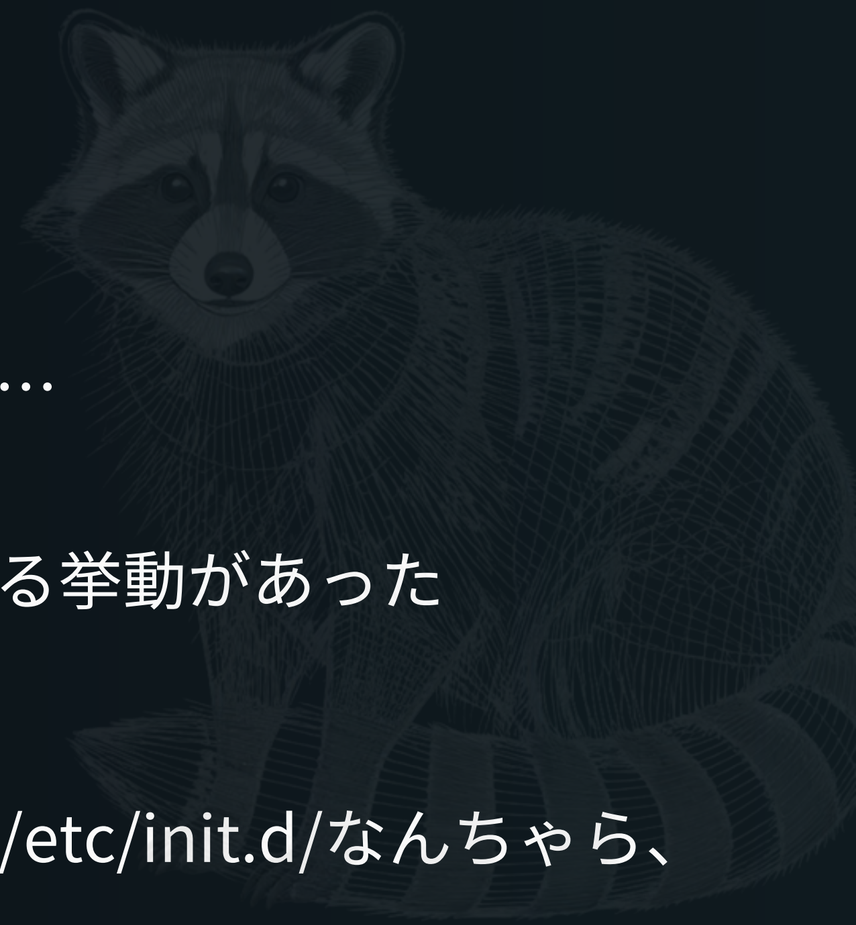
- これまでのようにNVIDIA, AMDのリポジトリへの依存がなくなるので管理的にはラクになる（それだけと言えばそれだけ）
- 「バグの報告先」がLaunchpadに統一というサブの価値はある
- どこまで＆いつまで追従されるのかは、今のところよく分からない

新機能のうちServerに影響しそうなもの：



- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

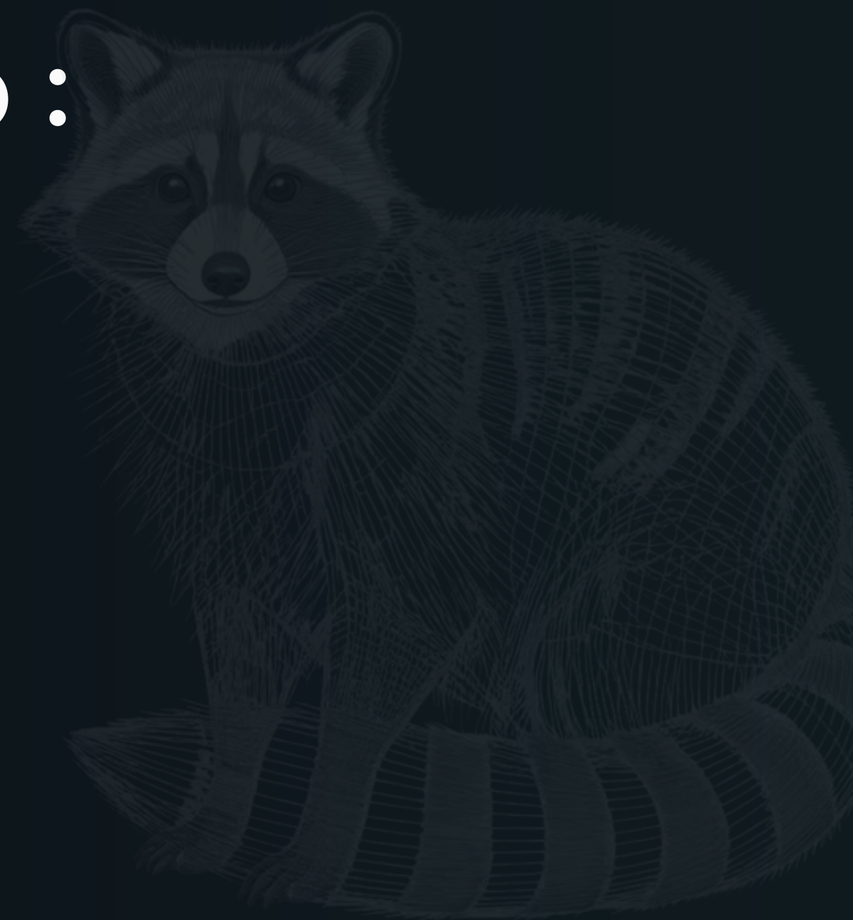
SysV Initスクリプト



- /etc/init.d/httpd start みたいなやつ
- systemdの存在によって過去のものになった……
  が、移植が完了しているとは言っていない
- systemdの機能として「読み替え」て動作させる挙動があった
- この挙動がなくなる
- パッケージ側は対処されるハズだが、自作した/etc/init.d/なんちゃら、
  のたぐいを移植する必要がある

※ アンタッチャブル状態になっている場合は頑張って移植しましょう……

新機能のうちServerに影響しそうなもの：



- x86-64v3（AMD64v3）の積極サポート
- coreutilsとsudoがRustに
- カーネル7.0（+ Intel TDX＆Enhanced IOMMUサポート開始）
- 仮想化＆コンテナスタックの更新
- 各種ソフトウェアのアップデート
- NTPdがChronyになった
- CUDA, ROCmがリポジトリに投入
- SysV initスクリプトへの後方互換性を持つ最後のリリース
- PQC対応への積極性

PQC（Post-Quantum Cryptography）

- 量子コンピューターによる暗号解読への、『一定の』安全性のある暗号系

  これまでは理論上の存在だったが最近AIの進展で俄然注目されている
  （AIが量子コンピューターで使えるアルゴリズムを作るとヤバい）

- 既存の暗号系は該当しない（素因数分解も離散対数も楕円曲線もダメ）

- クリプトアジリティ（暗号系の乗り換えやすさ）とセットで検討

- 暫定的な推奨
  （最終的に何になるかはまだ分からない）

26.04でのPQC Readyが謳われているが……

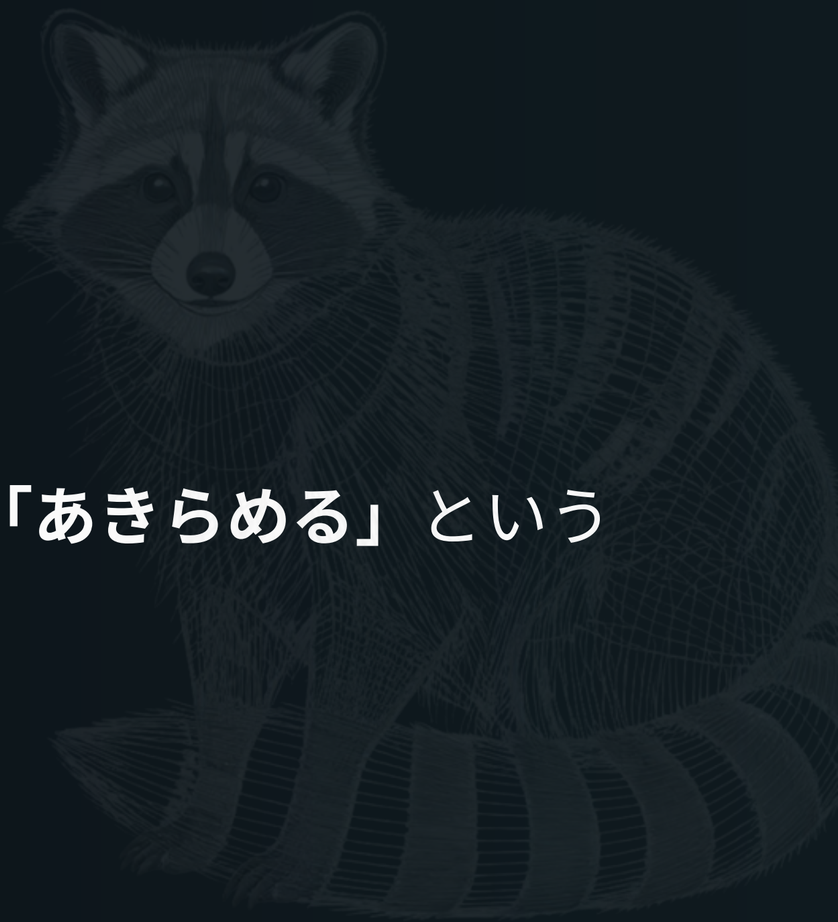
- PQCオンリーへの切り替えが可能、デフォルトはハイブリッド
- 26.04でなければムリ、という話ではない
  （各種アプリケーションをPPAで更新していけば古いバージョンでもOKではあるが、それはOKと言えるのか……？　みたいな問題はある）
- 将来的にFIPS-206で必須になるタイミングでは26.04が楽では？
  ぐらいの位置付け
- 28.04でどうなるかは、**今の暗号系がどれぐらい生きているか**による……

アップグレードする or しないの考え方

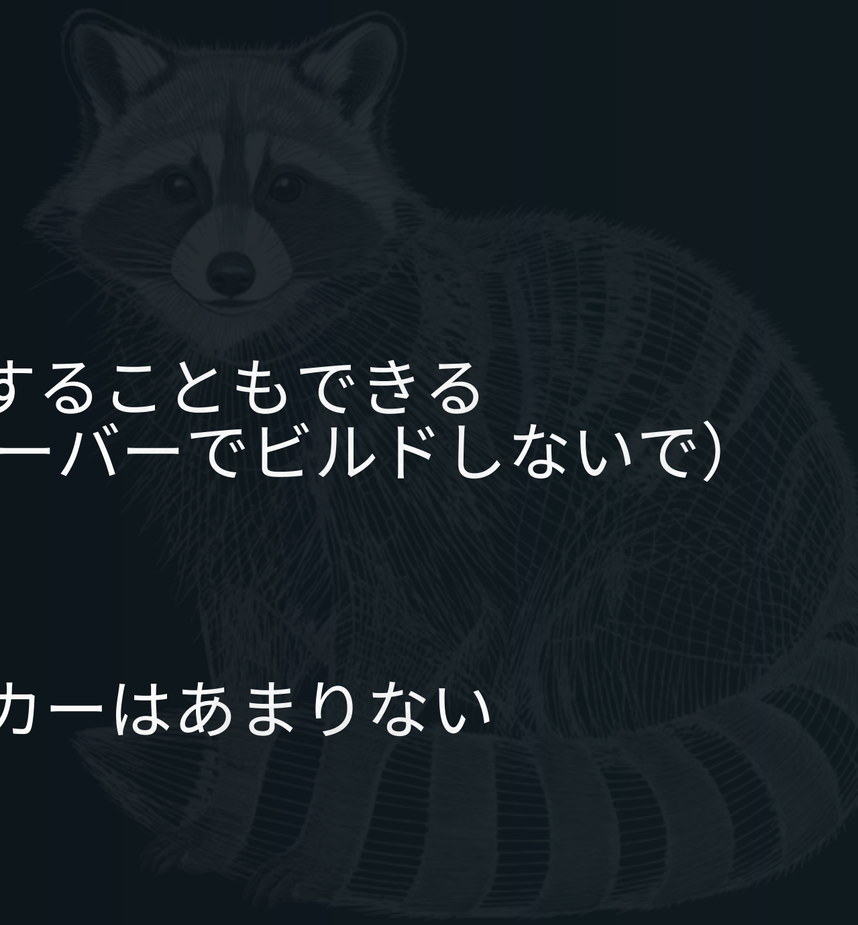
アプローチをどうするべきか

- **24.04 LTSには「見送る」という選択肢がある**
  （28.04 LTSが出てから考える、と同義）

- 22.04 LTSには**「26.04にする」「24.04にする」「あきらめる」**という選択肢がある。
  （あきらめる＝Ubuntu Proを調達する）

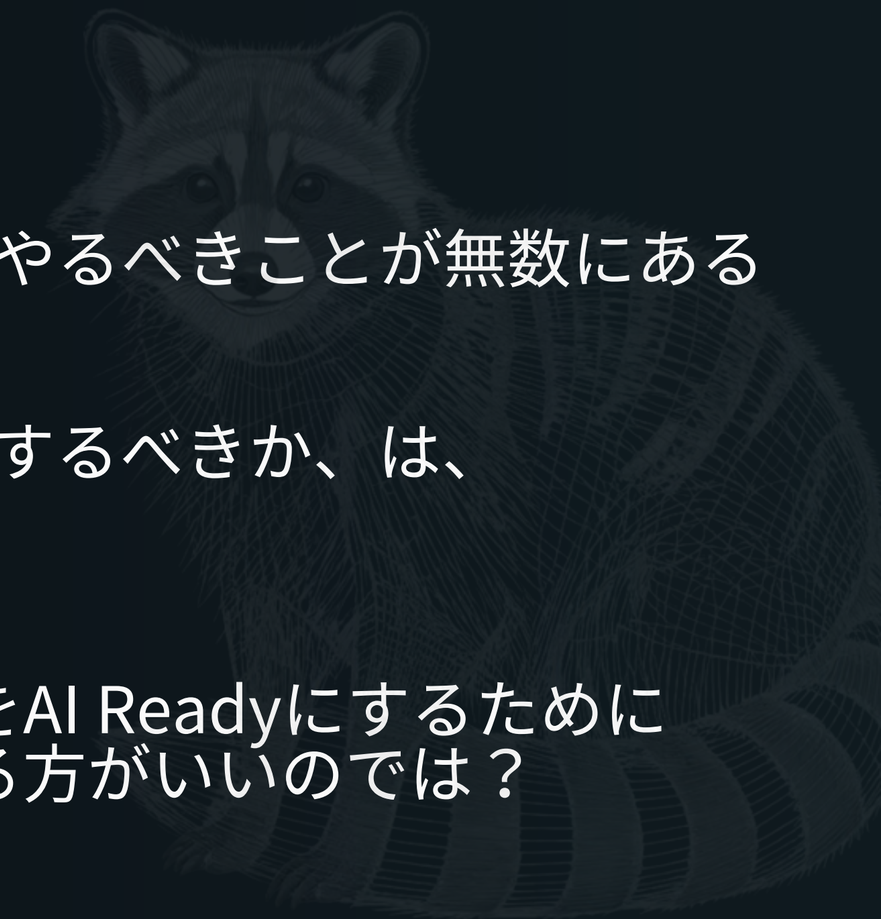


uutils（Rust版coreutils）コワイ問題



- たぶん怖くない（何も起きない）
- 何が起きたら or 踏みたくなければGNU実装にすることもできる
  （build-essentialsとの同居問題等はあるが、サーバーでビルドしないで）
- つまり、たぶん、あんまり問題にならない
  ≒ 26.04 LTSにすること「そのもの」のブロッカーはあまりない

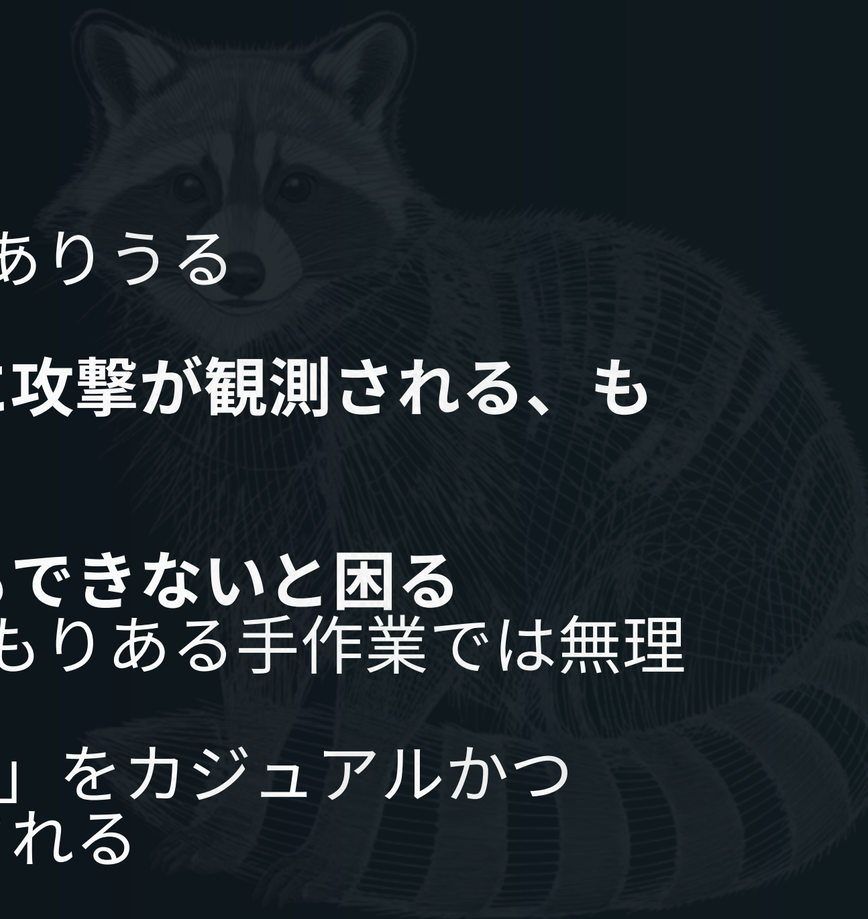
ただし「今」やるべきかというと



- 脆弱性対応やAI Readyな運用体制など、「今」やるべきことが無数にある
- 「アップグレードにかかる手間」をどこに投入するべきか、は、
  相当に悩む価値がある
- 今アップグレードするよりは、たとえば監視をAI Readyにするために
  メトリクスを見直したり、AI based SREを作る方がいいのでは？
- ただし、将来AIの性能は拡張されるはずなので、上記の投資もまた
  後回しにしてもいいのでは、という検討の対象
- セキュリティだけは待ったなし

今やるべきことの整理

セキュリティは待ったなし

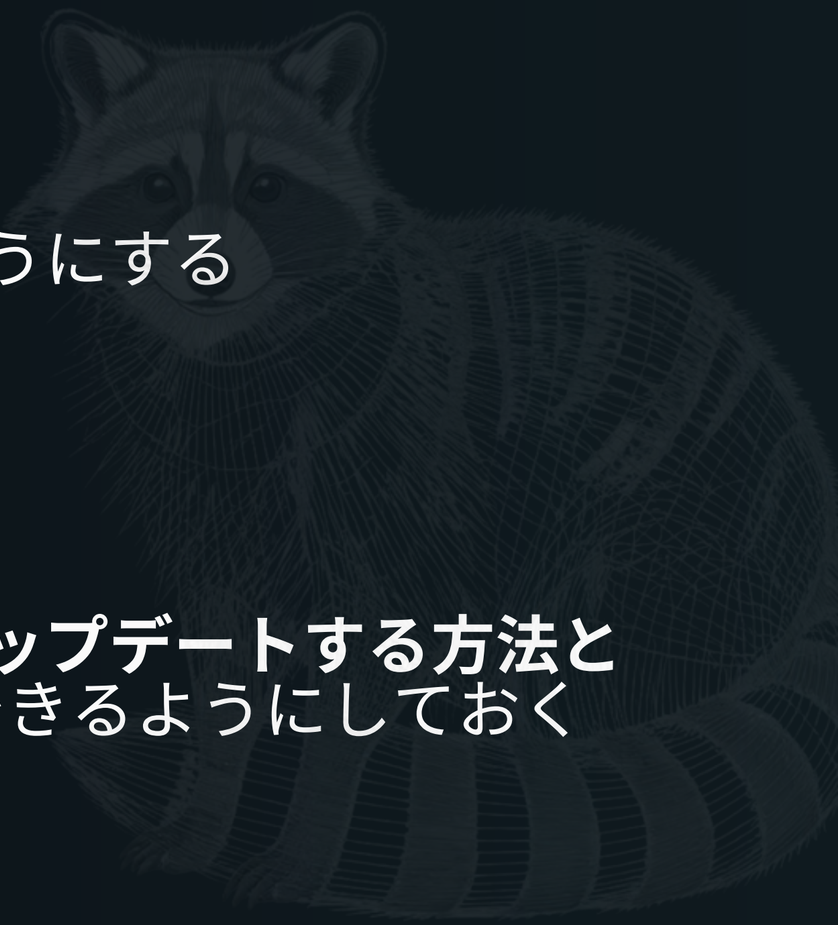


- 「毎日ヤバい脆弱性が出てくる」ぐらいまではありうる
- **アップデータが出てきたタイミングではすでに攻撃が観測される、も普通にありえる**
- **つまり「とにかく全部当て続ける」が最低でもできないと困る**
  「とにかく全部当てる」をするためには、ぬくもりある手作業では無理
- カーネルやglibcのような、「再起動を伴うもの」をカジュアルかつ高頻度に＆手間なく実施できること、が要求される

……という状態に到達させるのはおそらく最優先
これがまだならアップグレードを見送って先にやろう

具体的には？

- モニタリングで「なにか」起きたら気付けるようにする
  Runbookを準備する
- **SBOMを作る（作り続けられる状態にする）**
  EOL時期を整理する
- **システムを構成するソフトウェアを「全部」アップデートする方法と**
  「正常動作」を定義してAIエージェントに投入できるようにしておく
- クリプトアジリティの準備をする



あれ……だいぶヘビーでは……？

結論（再掲）：

- uutils由来の（Rust版）coreutilsはそこまでヤバくない
- 他の観点でもそこまでヤバくない
- ただ、他にやることがあるかもしれない